



# Data Management in STAR

---

Torre Wenaus

BNL

Persistent Data Workshop

11/13/98

# STAR Computing Requirements

Nominal year processing and data volume requirements:

Raw data volume: 200TB

Reconstruction: 2800 Si95 total CPU, 30TB DST data

- ◆ 10x event size reduction from raw to reco assumed (current: 3x)
- ◆ 1.5 reconstruction passes/event assumed

Analysis: 4000 Si95 total analysis CPU, 15TB micro-DST data

- ◆ 1-1000 Si95-sec/event per MB of DST depending on analysis
  - | Wide range, from CPU-limited to I/O limited
- ◆ ~100 active analyses, 5 passes per analysis
- ◆ micro-DST volumes from .1 to several TB

Simulation: 3300 Si95 total including reconstruction, 24TB

**Total nominal year data volume: 270TB**

**Total nominal year CPU: 10,000 Si95**



# STAR Computing Facilities

Dedicated RHIC computing center at BNL, the RHIC Computing Facility

- ◆ Charged with meeting the data archiving and processing needs (reconstruction and analysis; not simulation) of the four experiments
- ◆ Three production components: Reconstruction and analysis services (CRS, CAS) and managed data store (MDS)
- ◆ 10,000 (CRS) + 7,500 (CAS) Si95 CPU, balanced between CPU-intensive farm processing (reconstruction, some analysis) and I/O intensive SMP processing (I/O intensive analysis)
- ◆ ~50TB disk, 270TB robotic tape, 200MB/s, managed by HPSS

Limited resources require the most cost-effective computing possible

- ◆ Commodity Intel farms (running Linux) for all but I/O intensive analysis (Sun SMPs)

Support for (a subset of) physics analysis computing at home institutions



# Data Management at RHIC

RHIC Event Store Task Force last fall addressed data management alternatives

- ◆ Requirements formulated by the four experiments
- ◆ STAR and PHENIX selected Objectivity as the basis for data management
- ◆ ROOT selected by the smaller experiments and seen by all as analysis tool with great potential
- ◆ Issue for the two larger experiments:
  - | Where to draw a dividing line between Objectivity and ROOT in the data model and data processing

# Data Management Requirements

Support C++ with a well documented API.

Must scale to handle data set sizes at RHIC.

Ability to operate in range of the aggregate I/O activity required.

Currently support or plan to support an interface with HPSS

Provide adequate levels of integrity and availability.

Ability to recover from permanently lost data.

Object versioning and schema evolution.

Maintainable and upgradeable. Long-term availability.

Support for read/write access control at the level of individuals and groups.

Administration tools to manage the database.

Backup and recovery of subsets of the data.

Support for copying, moving data; data distribution over WAN

Control over data locality

# Objectivity in STAR

Objectivity selected for all database needs

- ◆ Event store
- ◆ Conditions and calibrations
- ◆ Online database and detector configurations

Decision to use Objectivity would have been inconceivable were it not for BaBar leading the way with an intensive development effort to deploy Objectivity on the same timescale as RHIC and at a similar scale

- ◆ STAR has *one* person (Jeff) on database development (event store, conditions and configuration); BaBar has 7-9

STAR has imported the full BaBar software distribution and deployed it as an ‘external library’ providing the foundation for STAR event store and conditions database

- ◆ BaBar’s software release tool and site customization features used to adapt their code without direct modification to the STAR environment

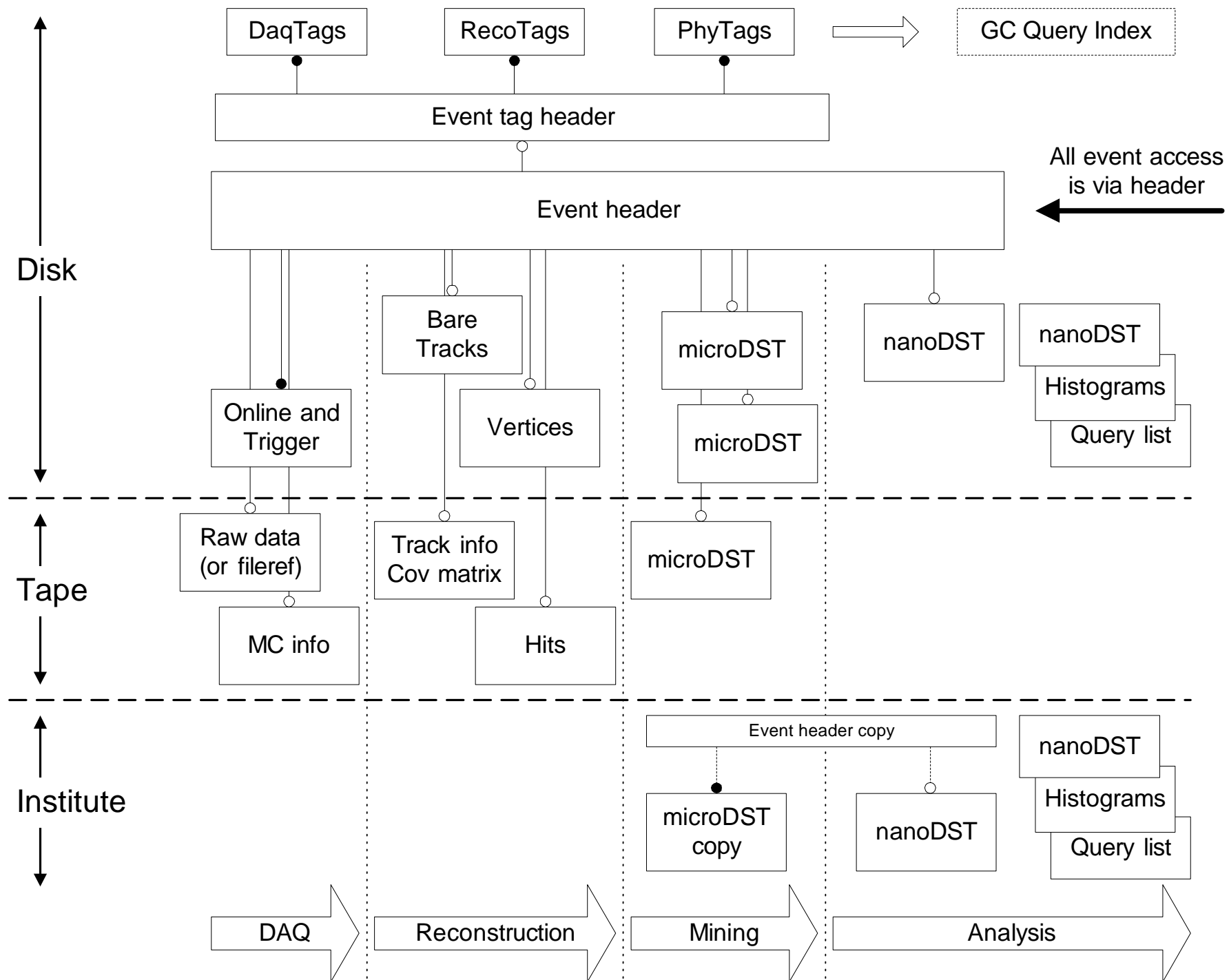
# Objectivity-based Event Store

BaBar event store software adapted to support STAR event store down to the level of event components

- ◆ Event collection dictionary with collection organization in Unix directory tree style
- ◆ System, group, user level authorization controls
- ◆ Federated database management: developer-level federations, schema evolution
- ◆ Event collection implementation
- ◆ Event components are purely STAR implementation and map directly to IDL-defined data structures

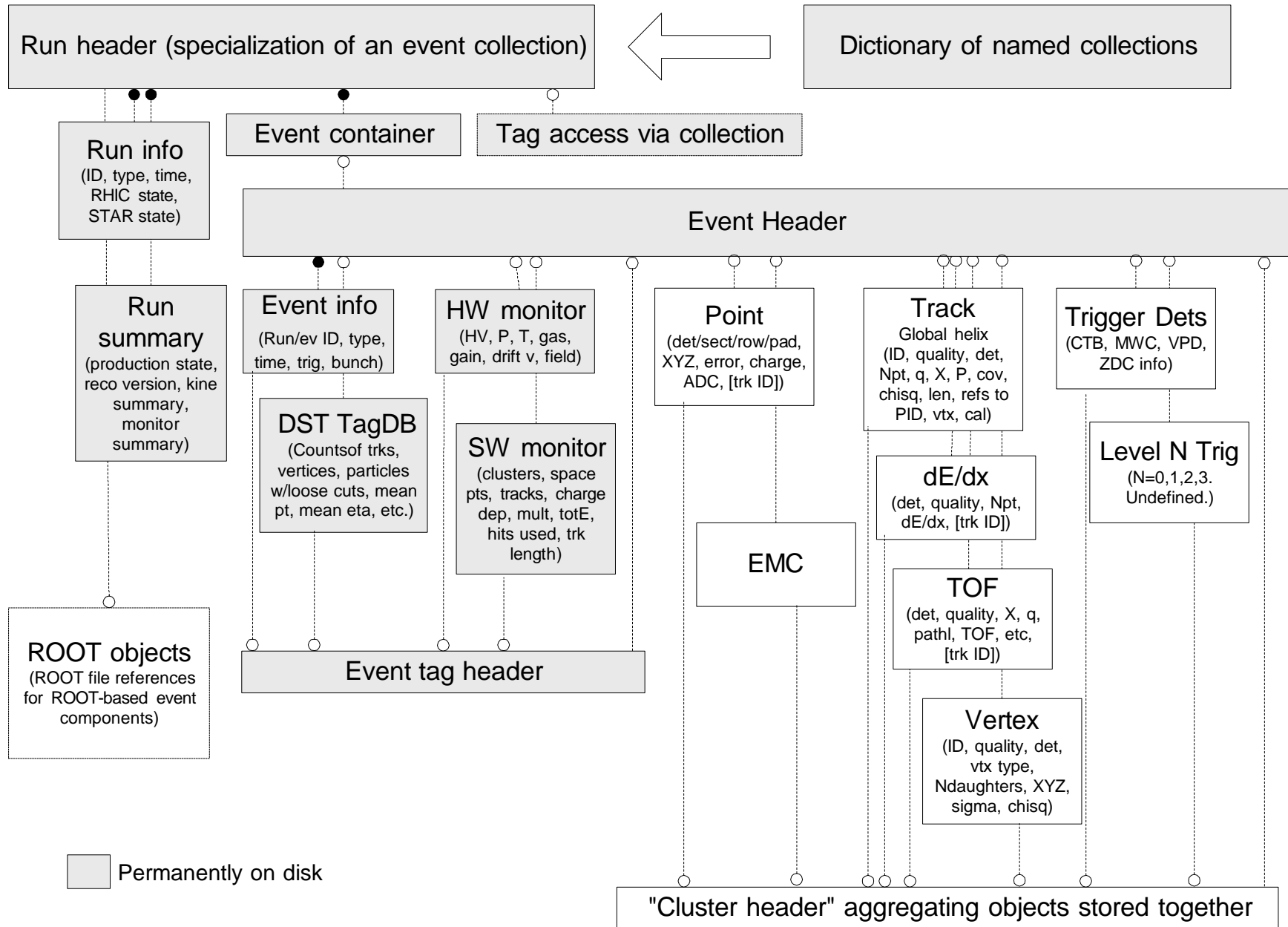
STAR raw data will **not** be stored in Objectivity

- ◆ In-house format will be used to insulate STAR against long-term viability of Objectivity-based storage
- ◆ Event store will provide file pointers to the raw data, and an Objectivity-wrapped version of the in-house format for small-scale convenience use



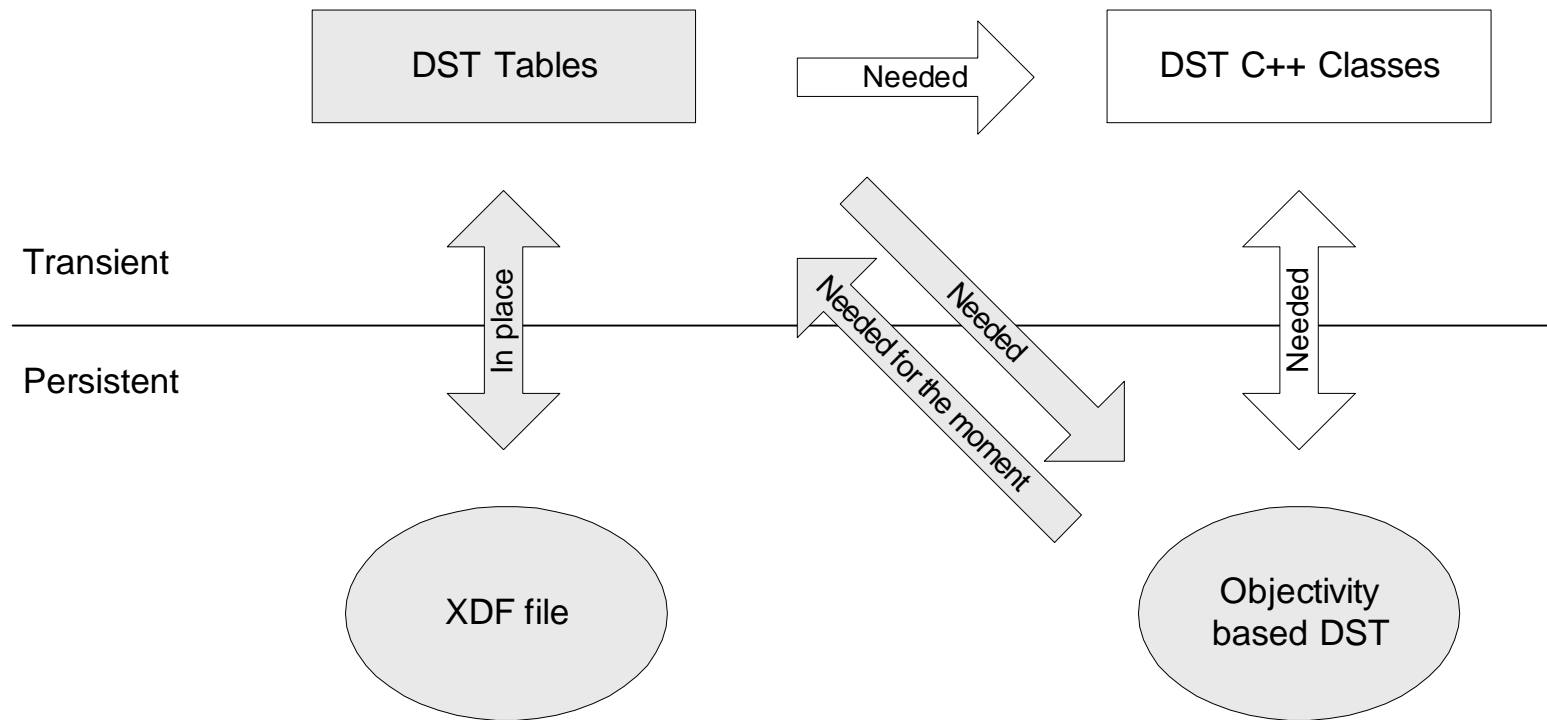




# Objectivity-Based Persistent DST Event Model for MDC1



# Persistent and Transient DST Event Representations

Objectives for MDC1



-  MDC1 baseline targets (met)
-  MDC1 target for secondary role (not met in MDC1)

# Conditions and Configuration Databases

Prototype shell (ie. empty) conditions database supporting time-dependent calibrations implemented based on BaBar's conditions database

- ◆ Like the event store, builds on existing IDL-based data model

Prototype configuration database supporting STAR's online system developed standalone, for later integration into BaBar framework

Jeff will address

# Data Access and Data Mining: Grand Challenge

Crucial issue in STAR is rapid, efficient data availability for physics analysis

- ◆ Full DST and post-DST volume well in excess of disk capacity

Grand Challenge addresses managed access to and ‘mining’ of the HPSS-resident Objectivity-based DST data

- ◆ Focused on RHIC and with heavy STAR involvement

Congrats to the GC on a successful test in MDC1 of a software apparatus

- ◆ very well focused on the priority needs of STAR and RHIC
- ◆ of immediate use to STAR for day to day operations
- ◆ on target to meet STAR’s needs for physics analysis

Thank you for building rapidly a very useful facility!

## Near term Use of GC Software

“Production” usage mode: independent activity at the physics analysis group level

- ◆ At the physicist level if we can get away with it (performance)

Management of the HPSS event store for STAF CAS users is the critical path item

- ◆ Retrieval by collection will be initial usage mode

Beyond that, would like to have available to users

- ◆ Query estimation and retrieval by query
- ◆ Retrieval by event list
- ◆ Logging, statistics, HPSS performance info
- ◆ Tagdb visualizer

Need direct API from TagDB to index builder, no intermediary file

- ◆ Simple ‘rebuild the index’ procedure

# Use of ROOT with GC

Assumption:

- ♦ ROOT usage of event store and ROOT usage of GC software are the same problem; when we solve the former we will have solved the latter
- ♦ No specific issue for GC

ROOT+Objy not touched yet. Lower priority than getting everything operating well under STAF.

# Event Store Implementation and Status

STAF-standard XDF format (based on XDR) for IDL-defined data is today the production standard and will remain an Objectivity fall-back

- ◆ IDL-defined persistent data model ensures XDF compatibility
- ◆ IDL-standard restricts how we use Objectivity: don't even pretend it's C++
- ◆ But, STAR (taking a lesson from BaBar) is completely decoupling persistent and transient data models, bearing the cost of translation
  - | Design of C++ transient model is decoupled from persistent implementation
  - | Transient representation is built in the translator
  - | No direct exposure of application code to Objectivity

Objectivity/BaBar based event store deployed in Mock Data Challenge and should soon be a production fixture of STAR data processing

Linux support is a crucial issue; currently limited to Sun/Solaris

- ◆ With Objectivity port available, BaBar/STAR software porting will proceed by end of year



# Database File Organization

Top-level metadata and tag files in \$STAR\_DB/stardb (sol disk)

- ◆ Event collections and headers
- ◆ Tag headers and tags

**Single directory** containing event data \$STAR\_DB/stardb/dst (disk1)

- ◆ Can live with single disk directory for the moment, but will need multi-directory capability soon (when we split event data into >1 cluster)
- ◆ Event database files correspond to reconstruction jobs with same name as XDF file

**HPSS migration** by moving event data files to /starreco/stardb/dst in HPSS

- ◆ **Compatible with GC?**

All DB loading so far done serially

- ◆ BaBar has seen problems in parallel loading; we need to test



# Event Store Database on Linux

Only operational platform for Objy event store at present is Sun/Solaris

No STAR Objectivity operations on Linux

- ◆ Biggest problem: BaBar/STAR database software port to Linux
  - | No attempt made yet to build on Linux
  - | At least some BaBar software reportedly needs egcs or 2.8.1, so we may have to wait for the next Objy/Linux version
  - | Rogue Wave reportedly does compile and work but not officially supported
  - | Will have a shot at building the BaBar essentials with gcc 2.7.3 and see whether encountered problems are surmountable
- ◆ If we get past that, until we get Objy version with Linux platform index overflow fixed, could build duplicate federations for Solaris and Linux

So, timescale for wanting to use GC from Linux unclear

# A Hybrid Event Store for STAR?

Requirements driving STAR to Objectivity are grounded in the very large scale data management problem

- ◆ So far (not very far) so good with Objectivity in the global data management role

Requirements and priorities for select components of the data model differ and can drive different choices

- ◆ Non-Objectivity raw data already addressed

Post-DST physics analysis data (micro-DSTs) is such an area

- ◆ High value in close coupling to data analysis tool, ROOT: direct access to data model during analysis, presentation
- ◆ Great physicist-level flexibility essential in defining object model (schema), and Objectivity (currently) presents severe problems in secure, flexible schema management
- ◆ Premium on storage optimization for compact, rapid-access data (compression, N-tuple like storage)

- ◆ More in Thomas' presentation



# Hybrid ROOT/Objectivity Event Store

These considerations motivate the use of ROOT for micro-DST level persistent data

- ◆ Particularly given the immaturity of the analysis tools being developed for LHC to work in conjunction with Objectivity-based data

Leads us to ROOT-based micro-DSTs integrated into the Objectivity event store

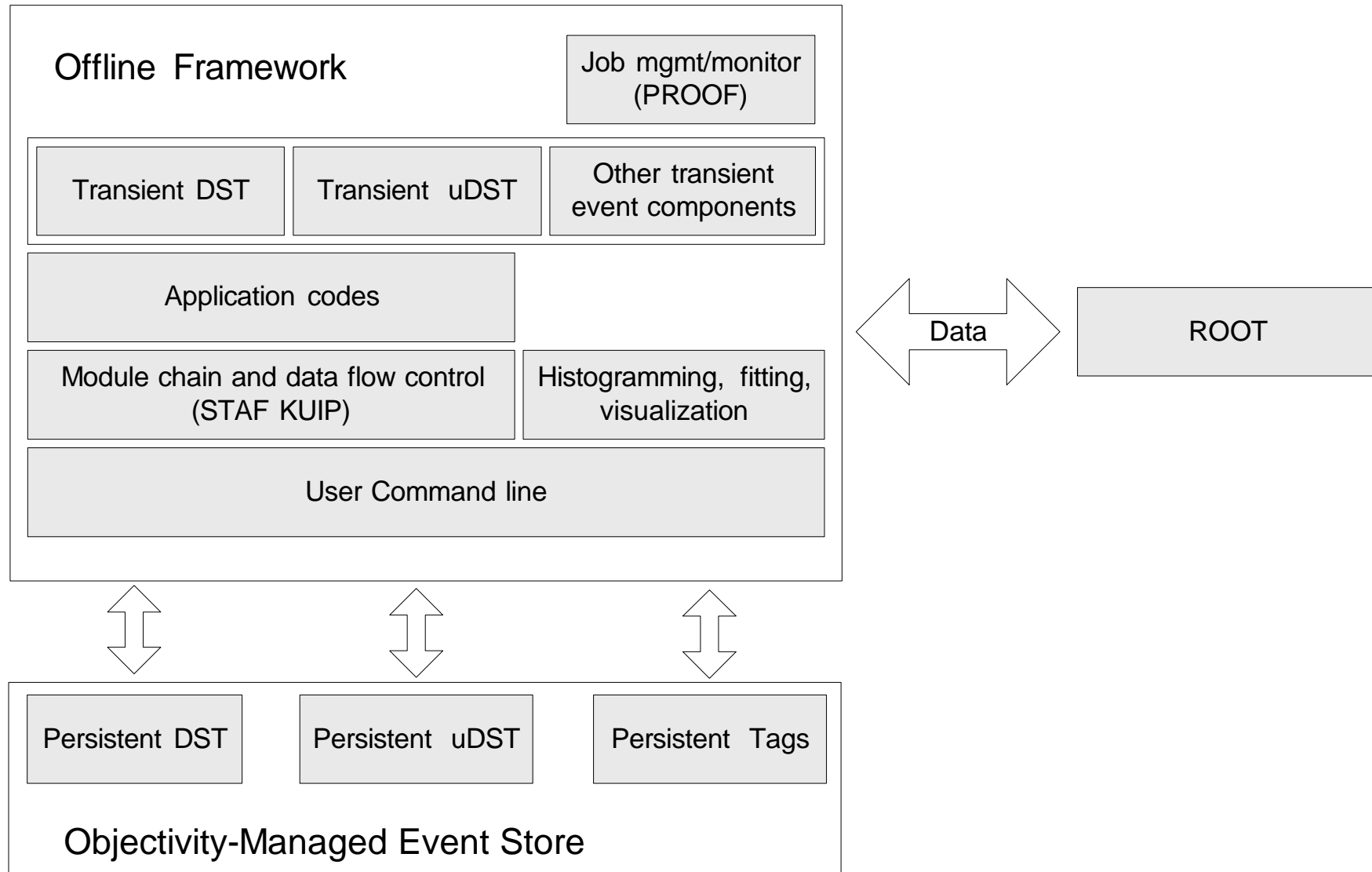
- ◆ ROOT-based micro-DSTs associated with an event collection
  - | ROOT file references at the collection level
- ◆ ROOT file contains event set corresponding to the collection
- ◆ Navigation from ROOT uDST entry to rest of event via Objy event header; always available, because navigation to the ROOT file is via an Objy collection

*Tags could be done exactly the same way; similar usage modes and optimization criteria to uDST*

# "Using ROOT" is many decisions, not one

Root can occupy or be used directly in any component...

... or none of them, used only as an external visualization/analysis tool



# MDC2 Wish List for GC

*(Preliminary)*

Multiple separately stored clusters per event

Retrieval of non-Objectivity components

- ◆ Objy Event store stores file reference, not the data itself
- ◆ We still pass an event ID and (set of) cluster ID's to be retrieved, but you recognize cluster ID's that refer to non-Objy files
- ◆ Uses:
  - | XDF fall-back for Objectivity!
  - | ROOT-format components
  - | Raw data retrieval

Retrieval of components from multiple COS's

Dynamic updating of index

Ability to specify (at administrator level) 'keep this data on disk'

- ◆ Ability to manage what material we keep on disk